

University of Applied Sciences Mittweida and Chemnitz University of Technology at TRECVID ActEv 2019

Rico Thomanek¹, Christian Roschke¹, Benny Platte¹, Tony Rolletschke¹, Tobias Schlosser², Manuel Heinzig¹, Matthias Vodel¹, Danny Kowerko², Frank Zimmer¹, Maximilian Eibl², and Marc Ritter¹

¹University of Applied Sciences Mittweida, D-09648 Mittweida, Germany

²Chemnitz University of Technology, D-09107 Chemnitz, Germany

Abstract. The analysis of video footage involving tasks such as identifying certain individuals at defined locations in a complex indoor or outdoor scenes or classifying person's activities still poses a challenge to any video retrieval system. Nowadays a variety of (semi-)automated analysis systems can be applied in order to solve some of its specific subproblems and the accuracy of object detection and simultaneously the detection and classification accuracy benefits strongly when latest cutting-edge machine learning methods such as deep-learning networks are involved. In this paper we propose our design of a heterogeneous video analysis system and report about our experiences with its application to the *Activity of Extended Video (ActEV)* analysis task within the *TREC Video Retrieval Evaluation (TRECVID)* contest. The proposed system improves the performance of person detection, identification and localization at predefined places in video scenes by heuristically combining a variety of state-of-the-art deep-learning frameworks for object detection and places classification into one heterogeneous system. The incorporated frameworks address a wide range of subproblems including stable object boundary extraction of salient regions or identifiable objects as well as person identification and object / place classification. Our approach integrates these processing artifacts using a feature-oriented approach in order to assess statistical correlations together with LSTM based activity classifiers across video frames.

1 Introduction to our Appearance at Activity in Extended Video

Over the last few years, the number of surveillance cameras has increased worldwide. Closed circuit television (CCTV) cameras record a steadily increasing amount of image data. Usually, such data is only reviewed after an specific event being thoroughly investigated for indications of noteworthy actions. In the domain of traffic safety (e.g. monitoring of intersections) or other sensitive areas, there is an increasing desire to evaluate the videos in a sensible and resource-saving way.

This paper focuses on our approach to solve the TRECVID Activity in Extended Video task (ActEv, [Awad et al. \(2019, 2018\)](#)) and in this case the automatic detection of object activities within surveillance areas. We build upon our work from the previous year (cf. to [Thomanek et al. \(2019, 2018\)](#)) and extend the proposed processing chains with complementary algorithms like long short-term memory networks.

Correspondence to: Marc Ritter
marc.ritter@hs-mittweida.de

2 System Architecture

Our system consists of several client units that perform various recognition tasks and use a database server for the persistent storage of all raw data and results. A distributed file system is used to access the stored raw data via HTTP, FTP, and SCP. It is thus not longer necessary to perform costly raw data processing steps repeatedly on various desktop computers. Instead all commonly used processing artifacts are stored directly in a database and made available to further high-level integration tasks via a suitable application protocol. For that purpose we developed an API that is able to provide any processing data in various exchange formats at application level. Our architecture implements a session management layer that allows for the addition or removal of processing nodes and services at any time during the process. The session management layer also facilitates scalability by distributing processing tasks to dynamically allocated resources.

The analysis process is fully automatized and continues until all allocated computing resources have been successfully completed. In error case, an error correction proce-

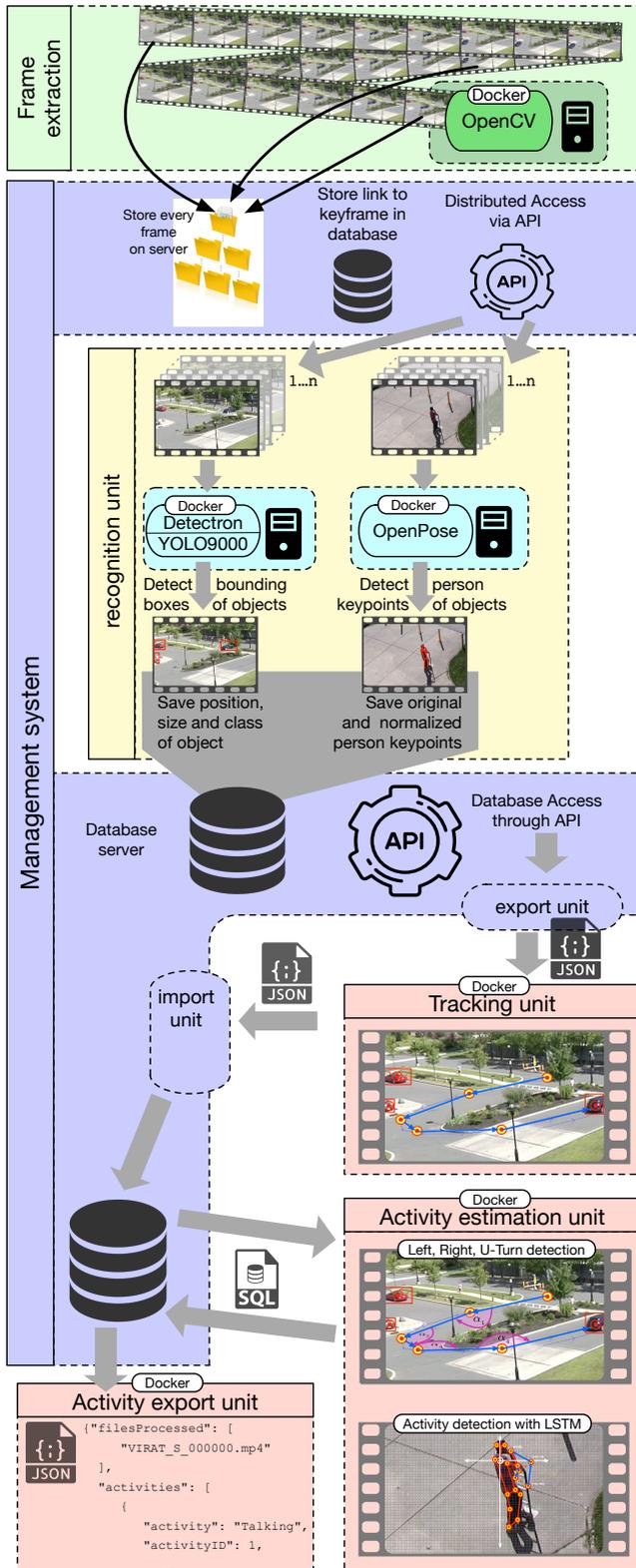


Figure 1: Our holistic system workflow for ActEV.

procedure is launched via the session management layer by triggering a data renewal process. After three unsuccessful attempts, data processing is terminated, marked as incorrect in the database, and the next processing block is provided. All intermediate and final processing results are then stored in the database and directly accessible to further high-level analysis clients based on the respective API. All services are executed as docker containers. This allows a fast deployment to distributed computing resources without the need to install additional software. The automatic resource management of (NVIDIA)-Docker also allows the containers to be executed multiple times. The number of processing instances is determined by CPU, RAM, and GPU memory utilization.

To solve the activities in Extended Video Task, as shown in Figure 1, open source frameworks for object identification and object tracking and a self-trained LSTM for activity detection were used. In a first processing step, the whole video corpus is split into frames, stored in the distributed file system and referenced in the database. Subsequently, a multi-person keypoint recognition as well as an object identification based on the extracted frames is performed. This extracted data is stored directly into the database, supplemented by additional standardized and transformed values created for the keypoints of detected persons. These values are normalized based on the distance between the keypoints of neck and hip being transformed into a local personal coordinate system whose zero point is placed at the neck. Based on this, the personal keypoints and recognized objects are stored in the database being transferred separately to a tracking algorithm. Each data set is successfully tracked over several frames and assigned a unique ID. The tracked person keypoints are then transferred to the long short-term memory CNN for activity determination; results are kept in the database. For the training of the LSTM, we developed a tool for the generation of synthetic human activities. The activity determination, based on the detected objects, is done by simple heuristics. In order to be able to deduce the specific activity, the extracted partial results are then merged using individual SQL queries.

In order to perform the ActEV task, the management system transfers all results received to the processing/scoring unit. This contains the business logic for evaluation and generates a result object, which is then transferred from the export unit to an XML or JSON container. The visualization unit uses these containers directly to visualize the tracking and activity detection results. This allows to create an interface for the intelligent annotation of the data for the ActEV task and to measure and evaluate the quality of the results intellectually after the competitive period for subsequent work or a more thorough analysis.

2.1 Frameworks

We use various state-of-the-art frameworks to recognize people and objects. Most of these frameworks only allow

directory-based processing of images. For the integration of such frameworks into the existing system environment, the source code needed to be extended by online processing functions. This means that the images to be processed no longer have to be stored locally on the host system, instead they are loaded directly from the central file storage during runtime. For this purpose a special API was developed, which enables parallel processing in addition to the provision of the data.

We separate the recognition of humans and objects from each other. All results are combined to feature vectors. Accordingly, the identification of an activity is achieved by making use of SQL queries. The relational linking of all results with the actual frame is done via the frame ID, which is realized via foreign keys. The following frameworks are used for a well-founded feature extraction: *OpenPose*, *Turi Create*, *Detectron*, and *Yolo9000*.

OpenPose is a library for the real-time recognition of person keypoints developed by *Hidalgo et al.* (Cao et al., 2018). There are now several implementations based on PyTorch, Keras or Tensorflow. OpenPose recognizes the human body and displays its key points on single images or enables their export in JSON format. In addition, the computing power of the system is invariant to the number of persons captured in the image for estimating the body key point. The used dataset is the CMU Panoptic Studio dataset, in which 480 synchronized video streams of several people involved in social activities were recorded and labeled, time-variable 3D structures of anatomical person key points in space were exported. The main functionality of the library is to estimate and interpret the body pose of persons using 15 or 18 key points. In addition, OpenPose provides the ability to estimate and render 2×21 hand key points. All functions are provided via an easy-to-use wrapper class.

TuriCreate is an open source framework for machine learning developed by Apple in 2017. The framework offers several libraries for the implementation of various machine learning tasks. These include e.g. activity determination based on sensor values, image classification, image similarity, sound classification, object detection, clustering or regression (Sridhar et al., 2018). Our use case for this framework focuses on activity detection. We use the framework to create an activity classifier using the keypoints from OpenPose as normalized and transformed sensor data. The basic model of the activity classifier in Turi Create is a deep learning model that can recognize temporal patterns in sensor data and is therefore well suited for the task of activity classification. The deep learning model is based on convolutional layers that extract temporal characteristics from a single prediction window, e.g. an arched arm movement might be an important indicator for telephoning. Furthermore, it relies on recurring layers to extract temporary features over time.

A framework that contains algorithms for object recognition is **Detectron**. It was developed by Facebook and is based on the Deep Learning Framework Caffe2 offering a

high quality and powerful source code collection for object recognition. Various algorithms like mask R-CNN, RetinaNet, Faster R-CNN, RPN, Fast R-CNN and R-FCN are implemented. In addition, multiple backbone network architectures such as ResNet50, ResNet101, ResNet152, VGG16 or Feature Pyramid Networks can be used. Facebook also offers a wide range of trained models for reuse with the *Detectron Model Zoo data set*. (Girshick et al., 2018).

According to our last years appearance, we use *Detectron* to retrieve objects and their boundary frames. In addition to the x and y coordinates of the starting point, we also store the height and width of the box in addition to the object classification. We also use ResNet152 as a backbone net in combination with Mask R-CNN. The classes assigned to each image are taken from the Coco dataset (Lin et al., 2014).

“YOLO” (You Only Look Once) is a network for real-time classification and localization of objects within an image. In contrast to a stack-based and step-by-step approach to object detection, object recognition is processed as a single regression problem. (Redmon et al., 2016)

YOLO9000 represents a revision of the YOLO framework, since it made significant mistakes, especially in object localization. Furthermore *YOLO9000* with over 9,000 different object categories was trained. (Redmon and Farhadi, 2016) We use *YOLO9000* to detect objects and their bounding boxes. The position and size of the bounding box is described by *YOLO9000* using upper left corner coordinates and lower right corner coordinates. The detected object class and the bounding box values are stored in the database.

2.2 Data handling and interface

For the permanent storage and provision of data we use the architecture described under (Thomanek et al., 2018).

As described, we use a chain of docker instances to process all data analysis tasks. Docker is open source software for isolating applications with container virtualization. Docker simplifies the deployment of applications by simplifying the transport and installation of containers with all necessary packages as files.

For video activity detection, it is necessary to analyze successive image data. This results in large amounts of image data that must be efficiently stored and distributed to the processing frameworks. To access this data, we have implemented a session management layer, which distributes the data to be processed to the frameworks working in parallel. For this purpose, each framework must log on to the session management and submit a processing request. Session management manages and monitors the processing process in the background and assigns unprocessed data to the free framework instances. The individual sessions are monitored using a decentralized heartbeat function, in which the framework instances must signal their availability at regular intervals. If an instance is no longer available due to errors, unprocessed data is then assigned to another free instance. The execu-

tion of multiple processing instances on physical hardware is determined by evaluating their CPU, GPU, and memory utilization. Depending on the resources required, the physical hardware can thus be optimally utilized and the processing time significantly reduced.

3 Workflow of Our Method

Our approach was to detect the required objects (vehicles, persons, ...) in the video material and to use their positioning and boundary box for activity detection. For this we used the frameworks *Detectron* and *YOLO9000*. In order to enable an activity determination as accurate as possible, each single frame was analyzed by means of *Detectron* and *YOLO9000*, which caused no performance losses compared to the procedure described in (Thomanek et al., 2018) due to the high parallelizability of the processes. To derive activities from the objects determined with *Detectron* and *YOLO9000*, their temporal relationship was recorded using a tracking algorithm and stored in the database with a unique tracking ID. For this purpose, the center of the boundary frames and the pattern information of all objects were used and their positional changes with relation to the previous and subsequent frames were determined. This approach was also applied to the *OpenPose* results.

To determine the activities “Vehicle_turning_left”, “Vehicle_turning_right” and “Vehicle_u_turn” the algorithm described under (?) was adapted, based on the definitions for activities and related objects described in (ActEV Team, 2019).

For the recognition of the activities “Entering”, “Riding”, “Talking”, “Pull” and “Exiting” simple heuristics were used, in which the temporal overlapping of the boundary frames of the involved objects is considered.

For the activities “specialized_talking_phone”, “specialized_texting_phone”, “Loading”, “Closing”, “Opening”, “Open_Trunk”, “Closing_Trunk”, “activity_carrying”, “Transport_HeavyCarry” and “Unloading” we implemented our own activity classifier that predicts the most probable activity based on the time-varying *OpenPose* keypoints. All detected activities were stored in the database and merged using SQL queries. For example, the activity “Open_Trunk” was only detected if there was a vehicle in the position range of the person.

3.1 Retrieval of Object Data for Tracking

We used two different tracking algorithms to track the detected persons and objects and merged the results stored in the database. These include a self-developed algorithm described in (Thomanek et al., 2019) and one described in (Agarwal and Suryavanshi, 2017), which we adapted for our system.



Figure 2: Selection of synthetically generated activity animations.

3.2 Activity detection using LSTM based activity classifier

To create a suitable ground truth for the activity classifier, we developed a “unity”-based tool that allowed us to create synthetic data for the activities we were looking for. This tool enabled the simultaneous recording of human activities from 24 different perspectives. Furthermore, it allowed to add slight variances to the animations so that multiple activity animations could be created. A total of 5535 synthetic animations were generated and decomposed into 536517 frames. An overview of the generated Ground Truth is shown in fig. 2.

Using *OpenPose* the body keypoints for the animated characters were then determined and integrated into our system environment. The body keypoints provided by “*OpenPose*” describe the positioning of the bone point within the analyzed image. In order to determine an activity from the point movement over several frames, it is necessary to transform these coordinates to a body-centered point. For this purpose we have chosen the neck as the origin of the coordinates. To perform the transformation, all body points associated with a person are reduced by the x- and y-coordinates of the neck point. Another problematic aspect is the image resolution of the source material, as the “*OpenPose*” results refer to it. In order to create a dimensionless activity classifier, all body points must also be normalized. We have defined the distance between neck and hip as one normalization range and scaled all other body points to the associated value range. fig. 3 illustrates this fact graphically. Based on the thus created normalized and transformed body keypoints, the activ-

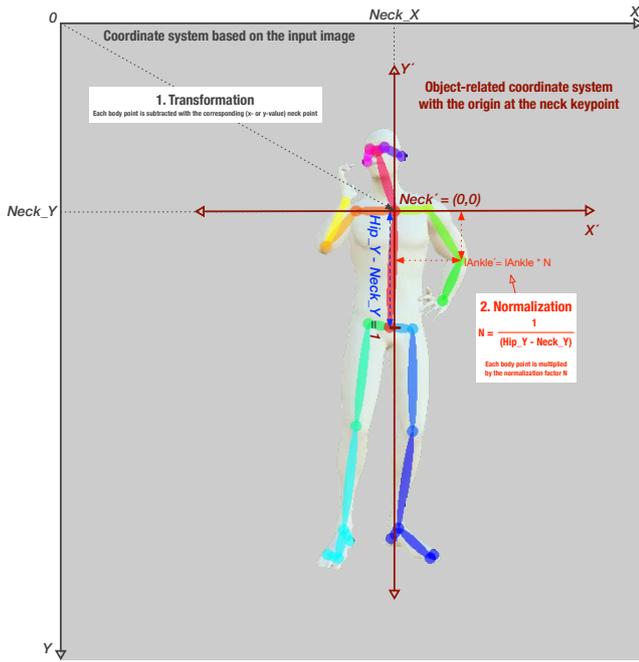


Figure 3: Normalization and transformation of the OpenPose results into an object-related coordinate system.

ity classifier was created using *Turi Create*. Each body point is regarded as a two-dimensional sensor. The COCO model used by us with *OpenPose* provides 18 body keypoints. Since each keypoint is described by an x- and y-coordinate, we get 36 individual sensor values. The activity classifier of *Turi Create* expects these sensor values, which are assigned to an activity, to be sorted chronologically in ascending order. The basic functionality of the activity classifier is shown in fig. 4.

The standardized and transformed “OpenPose” keypoints are then passed to the activity classifier to determine the activities to be searched for. The used prediction window was set to 15 frames. One highlight is the performance of the classifier, with a frame rate of 150 frames per second, which also allows real-time requirements. The detected activities are stored in the database with the probability value specified and then selected using threshold filtering. In combination with the objects detected by “Detectron” and “YOLO9000”, the most probable activity is concluded.

3.3 Activity detection using simple heuristics

Some activities are closely related to specific objects. Thus, the interaction of persons with certain objects (e.g. bikes, cars and bags) over a defined period of time can be associated with an activity. A basic requirement is an object detection for each individual frame, where all retrieved bounding boxes needs to be analyzed for intersections. For this purpose, a geometric function contained in PostgreSQL is used to calculate intersection values directly in the database. To ensure that it is not just a short touch or noise, all common

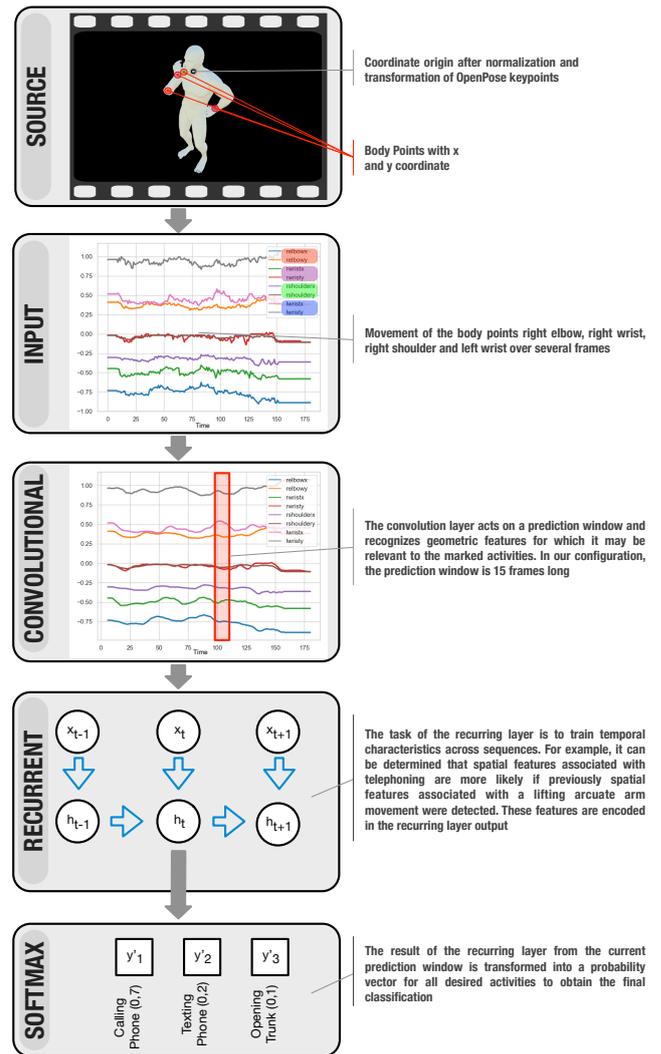


Figure 4: Basic functionality of the activity classifier.

frames of the interacting objects are determined and interpolated if necessary. This results in a common start and end frame and an intersection value for each object. The individual objects are identified by considering the tracking results. In addition, the direction vectors of the tracked objects are determined and included into the evaluation. This allows us to check whether the objects are separating, moving towards each other or even moving along the same direction. However, if a tracking algorithm fails or produces incorrect results, this can lead to negative effects on the activity detection. Therefore, it can happen that the same object within a video disappears for a short time and reappears later with a different ID. This leads to incomplete activity recognition results. We have tried to solve this problem with the help of a clustering approach.

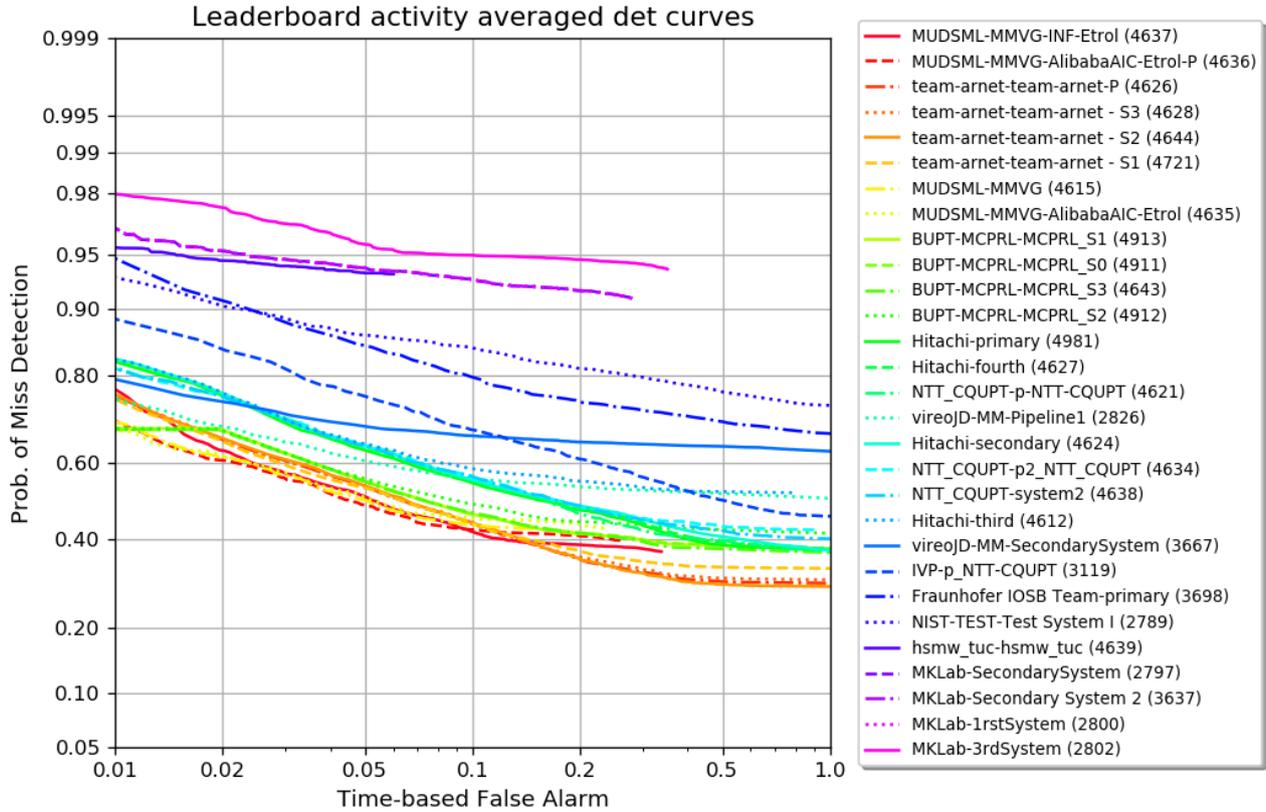


Figure 5: Results of the ActEV leaderboard evaluation (status 24th of October 2019) showing our approach at *hsmw_tuc-hsmw_tuc* (4639).

4 Results and Future Work in Activity Event Detection

The results of our approach are shown in Figure 5. With mediocre results at the normalized partial area under the detection error tradeoff curve NIST (2018) is 0.94064. Furthermore, our system achieves an average miss rate of the same magnitude of 0.93551 at the time-based false alarm rate of 0.15 as well as 0.94371 at the same rate of false alarms.

Detection and tracking are still performed on the 2D projection plane of the video footage still neglecting any perspective corrections. We positively tested the proposed infrastructure on the given tasks. However, there is still a lot of space for improvements with much more sophisticated and elaborated methods.

Acknowledgments

The European Union and the European Social Fund for Germany partially funded this research. This work was also partially funded by the German Federal Ministry of Education and Research in the program of Entrepreneurial Regions InnoProfileTransfer in the project group localizeIT (funding code 03IPT608X). Program material in sections 1–3 is copy-

righted by BBC. We want to thank all the organizers of these tasks, especially George Awad and Afzal Godil, for the hard work they put into the annotation, evaluation and organization of these challenges.

References

- ActEV Team, N.: Draft of TRECVID2019 ActEV Evaluation Plan, 2019.
- Agarwal, A. and Suryavanshi, S.: Real-Time* Multiple Object Tracking (MOT) for Autonomous Navigation, Report, <http://cs231n.stanford.edu/reports/2017/pdfs/630.pdf>, 2017.
- Awad, G., Butt, A., Curtis, K., Lee, Y. L., Fiscus, J., Godil, A., Delgado, A., Smeaton, A. F., Graham, Y., Kraaij, W., Quénot, G., Magalhaes, J., Semedo, D., and Blasi, S.: TRECVID 2018: Benchmarking Video Activity Detection, Video Captioning and Matching, Video Storytelling Linking and Video Search, in: Proceedings of TRECVID 2018, NIST, USA, 2018.
- Awad, G., Butt, A., Curtis, K., Lee, Y., Fiscus, J., Godil, A., Delgado, A., Smeaton, A. F., Graham, Y., Kraaij, W., and Quénot, G.: TRECVID 2019: An evaluation campaign to benchmark Video Activity Detection, Video Captioning and Matching, and Video Search & retrieval, in: Proceedings of TRECVID 2019, NIST, USA, 2019.

- Cao, Z., Hidalgo, G., Simon, T., Wei, S.-E., and Sheikh, Y.: Open-Pose: realtime multi-person 2D pose estimation using Part Affinity Fields, in: arXiv preprint arXiv:1812.08008, 2018.
- Girshick, R., Radosavovic, I., Gkioxari, G., Dollár, P., and He, K.: Detectron, <https://github.com/facebookresearch/detectron>, 2018.
- Lin, T.-Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C. L., and Dollár, P.: Microsoft COCO: Common Objects in Context, ArXiv e-prints, 2014.
- NIST: Draft of ActEV 2019 Evaluation Plan, https://actev.nist.gov/pub/ActEV_TRECVID_EvaluationPlan_081219.pdf, viewed: 2019-10-24, 2018.
- Redmon, J. and Farhadi, A.: YOLO9000: Better, Faster, Stronger, arXiv.org, p. arXiv:1612.08242, <http://arxiv.org/abs/1612.08242v1>, 2016.
- Redmon, J., Divvala, S. K., Girshick, R. B., and Farhadi, A.: You Only Look Once - Unified, Real-Time Object Detection., CVPR, pp. 779–788, doi:10.1109/CVPR.2016.91, <http://ieeexplore.ieee.org/document/7780460/>, 2016.
- Sridhar, K., Larsson, G., Nation, Z., Roseman, T., Chhabra, S., Giloh, I., de Oliveira Carvalho, E. F., Joshi, S., Jong, N., Idrissi, M., and Gnanachandran, A.: Turi Create, <https://github.com/apple/turicreate>, viewed: 2018-10-12, 2018.
- Thomanek, R., Roschke, C., Manthey, R., Platte, B., Rolletschke, T., Heinzig, M., Vodel, M., Kowerko, D., Kahl, S., Zimmer, F., Eibl, M., and Ritter, M.: University of Applied Sciences Mitweida and Chemnitz University of Technology at TRECVID 2018, Gaithersburg, Maryland, USA, 2018.
- Thomanek, R., Roschke, C., Platte, B., Manthey, R., Rolletschke, T., Heinzig, M., Vodel, M., Zimmer, F., and Eibl, M.: A scalable system architecture for activity detection with simple heuristics, in: Proceedings - 2019 IEEE Winter Conference on Applications of Computer Vision Workshops, WACVW 2019, doi:10.1109/WACVW.2019.00012, 2019.